**Amendments to the Specification**

Please replace the paragraph bridging pages 5 and 6 with the following amended paragraph:

The above problem can be resolved and the object can be achieved, if in a complex computerized system having a processor for processing commands, and one or more shared resources (such as a number for memory units, databases, hardware units and the like) required for executing said commands, there is applied a method for effective ~~utilizing~~ utilization of the shared resources (at a command level), and the method comprises steps of:

- deriving, from each of said commands, subcommands respectively related to said one or more shared resources,
- assigning priorities to said subcommands,
- forwarding said subcommands to one or more input queues of the respective one or more shared resources, so that each of said input queues comprises the subcommands related to a particular shared resource, thereby ensuring execution of the subcommands from said queues by said shared resources in an asynchronous manner, and according to said subcommand priorities by each of the shared resources.

Please replace the paragraph bridging pages 6 and 7 with the following amended paragraph:

In a typical case, where there are many shared resources and more than one command to be executed, the method provides for simultaneous ~~utilizing~~ utilization of said different shared resources and consequently, for asynchronous execution of the commands. In such a typical case, the list of shared resources required for execution of one command, at least partially overlap the list of shared resources required for execution of another command. However, even in a case where only one command is to be currently executed using one shared resource, the method still applies since this command may comprise a number of subcommands requiring utilizing of the shared resources but having "different importance". Even in such a case, assigning priorities to the subcommands would allow using the shared resource effectively, since a more urgent operation or a newly arriving subcommand of another command, would have a chance to be executed by the shared resource without waiting for the prior command completion.

Please replace the first full paragraph on page 10 with the following amended paragraph:

Preferably, the above control system for effective utilizing shared resources of a computerized system further comprises a higher level ~~(master)~~ processor capable of

cooperating with said ~~(lower level or slave)~~ command processors having a level lower than said higher level processors; said higher level processor being operative to distribute commands between said command processors, and receive from said command processors reports to respective commands.

Please replace the third paragraph on page 12 with the following amended paragraph:

**Fig. 1** schematically illustrates a control system 10 configured for effectively utilizing shared resources of a computerized system at a command level. The control system comprises a central processor 12 (a higher level processor) connected to "n" command processors (lower level processors). Only the first and the "n"-th command processors are shown in the drawing and marked 14 and 16, respectively. The central processor 12 receives user's external requests 11 (commands), and issues replies 13 being solutions to the problems which the requests set. For example, the request may constitute an urgent requirement to reroute and restore a particular telephone call when its original path in the network failed.

Please replace the last paragraph on page 14 with the following amended paragraph:

It is therefore understood, that the shared resources operate in parallel and in the most effective way, i.e., each of the shared resources is not locked for the complete time of execution of any command, but devotes to any command only the exact time required for execution one or more of its subcommands. Therefore, time which was wasted in the prior art solutions due to locking a shared resource during one command, now remains available for serving other commands in the asynchronous manner by allowing at least one subcommand of one command to start executing while subcommands of another command are not finished executing.